EXAMINER'S AMENDMENT

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on October 20/2009 has been entered.

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

3. Authorization for this examiner's amendment was given in a telephone interview with William J. Cooper (Reg. No. 44,629) on November 6, 2009.

4. Amend the claims as follows:

1. (Currently Amended) A computer-readable storage medium having computer-executable instructions, that when executed on a computing system, perform a method comprising:

implementing an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and outputting demultiplexed elementary media streams, the interface comprising:

an Initialize method configuring the demultiplexer object;

a GetPendingPresentationDescriptor method retrieving the next pending presentation;

a SetPresentationDescriptor method dynamically setting an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists only ~~merely~~ if all output associated with the active presentation has been serviced, wherein the active presentation descriptor describes media types of an elementary stream, and facilitates selection of streams to be extracted by the demultiplexer object;

wherein if the SetPresentationDescriptor method is called attempting to set the active ~~presention~~ presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, wherein the

SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method providing a new input muxed stream to the demultiplexer object;

a ProcessOutput method retrieving at least one elementary stream from an active presentation determined based on the dynamically set active presentation descriptor; and

a Flush method flushing currently queued input and output samples.

2. (Currently Amended) The computer-readable storage medium of claim 1 wherein the interface further comprises a GetPresentationDescriptor method ~~to retrieve~~ retrieving a clone of the currently active presentation descriptor on the demultiplexer object.

3. (Previously Presented) The computer-readable storage medium of claim 2 wherein the GetPresentationDescriptor method includes a presentation descriptor.

4. (Cancelled).

5. (Currently Amended) The computer-readable storage medium of claim [[4]] 1 wherein the GetPendingPresentationDescriptor method includes a pending presentation descriptor.

6. (Previously Presented) The computer-readable storage medium of claim 1 wherein the Initialize method includes parameters, the parameters comprising:

a muxed stream descriptor;

a selected media type for the muxed stream descriptor;

an array of major types of elementary streams;

and a count of major types in the array of major types.

7. (Currently Amended) A computer implemented method ~~for~~ comprising:

implementing, using a processor, an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and ~~outputting~~ outputting demultiplexed elementary media streams, the interface comprising:

an Initialize method configuring the demultiplexer object;

a GetPendingPresentationDescriptor method retrieving the next pending presentation;

a SetPresentationDescriptor method dynamically setting an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists only ~~merely~~ if all output associated with the active presentation has been serviced, wherein the active presentation descriptor describes media types of an elementary stream, and facilitates selection of streams to be extracted by the demultiplexer object;

wherein if the SetPresentationDescriptor method is called attempting to set the active ~~presention~~ presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method providing a new input muxed stream to the demultiplexer object, wherein the ProcessInput method includes a pointer to a sample object;

a ProcessOutput method retrieving at least one elementary stream from an active presentation determined based on the dynamically set active presentation descriptor, wherein the ProcessOutput method includes a stream identifier and a pointer to a pointer to a sample object; and

a Flush method flushing currently queued input and output samples.

8. (Previously Presented) The computer-readable storage medium of claim 1 wherein the ProcessInput method includes a pointer to a sample object.

9. (Previously Presented) The computer-readable storage medium of claim 8 wherein the ProcessInput method includes a return value having a new presentation flag.

10. (Currently Amended) The computer-readable storage medium of claim 9, having further computer executable instructions for performing the method comprising:

if the new presentation flag has a TRUE value:

calling a GetPendingPresentationDescriptor method to retrieve the next pending presentation;

selecting desired streams; and

calling the SetPresentationDescriptor method to enable processing of samples from the demultiplexer's input queue.

11. (Previously Presented) The computer-readable storage medium of claim 1 wherein the ProcessOutput method includes a stream identifier and a pointer to a pointer to a sample object.

12. (Previously Presented) The computer-readable storage medium of claim 11 wherein the ProcessOutput method includes an output return value.

13. (Previously Presented) The computer-readable storage medium of claim 12 wherein the output return value includes one of an end of stream error code and a no more data error code.

14. (Currently Amended) A computer-readable storage medium having computer-executable instructions, that when executed on a computing system, perform a method comprising:

implementing an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and outputting demultiplexed elementary media streams, wherein the interface takes multiplexed data as an in-memory buffer of data, the interface comprising:

an Initialize method configuring the demultiplexer object;

a GetPendingPresentationDescriptor method to retrieve the next pending presentation;

a SetPresentationDescriptor method dynamically setting an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists ~~merely~~ only if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presention descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method providing a new input muxed stream to the demultiplexer object, wherein the ProcessInput method includes a return value having a new presentation flag;

if the new presentation flag has a TRUE value:

calling a GetPendingPresentationDescriptor method to retrieve the next pending presentation;

selecting desired streams; and

calling the SetPresentationDescriptor method to enable processing of samples from the demultiplexer's input queue;

a ProcessOutput method retrieving at least one elementary stream from an active presentation determined based on the dynamically set active presentation descriptor;

a Flush method flushing currently queued input and output samples; and

a GetPresentationDescriptor method to retrieve a clone of the currently active presentation descriptor on the demultiplexer object~~; and~~

~~a GetPendingPresentationDescriptor method to retrieve the next pending presentation;~~

~~if the new presentation flag has a TRUE value:~~

~~calling a GetPendingPresentationDescriptor method to retrieve the next pending presentation;~~

~~selecting desired streams; and~~

~~calling the SetPresentationDescriptor method to enable processing of samples from the demultiplexer's input queue~~.

15. (Previously Presented) The computer-readable storage medium of claim 14 wherein the multiplexed data has a format comprising at least one of Digital Video, MPEG2, and ASF.

16. (Currently Amended) The computer-readable storage medium of claim 1, having further computer executable instructions for performing the method comprising:

storing an Initialize data structure for use in a demultiplexer, comprising:

a first field containing a header; a second field containing a muxed stream descriptor;

a third field containing a selected media type of the muxed stream descriptor;

a fourth field containing an array of major types of elementary streams; and

a fifth field containing a count of major types in the array of major types.

17. (Currently Amended) The computer-readable storage medium of claim 1, having further computer executable instructions for performing the ~~steps~~ method comprising:

storing a SetPresentationDescriptor data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a presentation descriptor.

18. (Currently Amended) The computer-readable storage medium of claim 1, having further computer executable instructions for performing the method comprising:

storing a GetPresentationDescriptor data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a presentation descriptor.

19. (Currently Amended) The computer-readable storage medium of claim 1, having further computer executable instructions for performing the method comprising:

storing a GetPendingPresentationDescriptor data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a pending presentation descriptor.

20. (Currently Amended) The computer-readable storage medium of claim 1, having further computer executable instructions for performing the method comprising:

storing a ProcessInput data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a pointer to a sample object.

21. (Currently Amended) The computer-readable storage medium of claim 1, having further computer executable instructions for performing the method comprising:

storing a ProcessOutput data structure for use in a demultiplexer, comprising:

a first field containing a header;

a second field containing a stream identifier; and

a third field containing a pointer to a point to a sample object.

5. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

CONCLUSION

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

7. U.S. Patent 6,891,547 B2 to Kang et al. discloses a decoding apparatus to decode multimedia data into original signals of respective types.

8. U.S. Patent 7,603,024 B1 to Chun discloses decoding a transport stream (TS) packet and reads additional information from program specific information (PSI).

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KimbleAnn Verdi whose telephone number is (571)270-1654. The examiner can normally be reached on Monday-Friday 7:30am-5:00pm EST.

10. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on (571) 272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

11. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

November 6, 2009
KV

/VAN H NGUYEN/
Primary Examiner, Art Unit 2194